

## **Formål**

Hovedformålet med denne øvelse er at anvende instruktioner til digital I/O.

Desuden introduceres praktisk anvendelse af PWM signaler (intensitetsstyring af lysdiode).

## **Materiale**

Lærebogen (Mazidi):

Afsnit 3.2. "Call instructions and stack".

Kapitel 4. "AVR I/O programming".

Lektion 7: "Stack".

Lektion 8: "Digitale porte".

## **Øvelsen**

I denne øvelse vil vi:

Del 1:

- Arbejde videre med løsningen til LAB-øvelse 3, således at "Mega2560 I/O shield"s 8 trykknapper kan anvendes til at spille toner i en C dur skala.

Del 2:

- Implementere en lysdæmper for en af "Mega2560 I/O shield"s lysdioder, således at lysintensiteten kan styres via trykknapperne.

### Del 1 ("Klaver")

I LAB-øvelse 3 lavede vi et program, der anvendte en lyd giver til at spille en C dur skala.

Vi vil nu arbejde videre med denne øvelse. En mulig løsning på LAB3 kunne være følgende:

```
;***** INITIERING *****
;**** Klargoer SP og PORTB ****
    LDI R16,HIGH(RAMEND) ;Klargoer Stack Pointer
    OUT SPH,R16
    LDI R16,LOW(RAMEND)
    OUT SPL,R16
    SER R16                ;PORT B er udgang
    OUT DDRB,R16
    CLR R16                ;Sluk alle lysdioder
    OUT PORTB,R16

;***** PROGRAM LOOP *****
    LDI R20,239    ;c
    CALL TONE
    LDI R20,213    ;D
    CALL TONE
    LDI R20,190    ;E
    CALL TONE
    LDI R20,179    ;F
    CALL TONE
    LDI R20,160    ;G
    CALL TONE
    LDI R20,142    ;A
    CALL TONE
    LDI R20,127    ;H
    CALL TONE
    LDI R20,120    ;C
    CALL TONE
HERE:
    JMP HERE        ;Bliv her "altid"

;*****
;* DELAY (R18*4us)      *
;* Laver en tidsforsinkelse på *
;* "Det der er i R18" gange 4 us *
;*****
DELAY:
    LDI R17,20    ;<-- OPGAVE (del 1): Find xxx og skriv det her
AGAIN:
    DEC R17
    BRNE AGAIN
    DEC R18
    BRNE DELAY
    RET

;*****
;***** SPILLER EN TONE *****
;***** 250 HALVPERIODER *****
;***** T/2 = (R20)*4 us *****
;*****
TONE:
    LDI R19,250    ;Tonen skal vare 250 halvperioder
IGEN:
    COM R16        ;Toggle (inverter) alle ben på PORTB
    OUT PORTB,R16
    MOV R18,R20    ;Kopier R20 til R18
    CALL DELAY     ;Vent 1/2 periodetid
    DEC R19
    BRNE IGEN      ;Gentag 250 gange
    RET            ;Retur til program loop
```

Der skal nu laves ændringer i programmet, således at programmet ikke længere "automatisk" spiller de 8 toner, men vi ønsker i stedet at kunne anvende de 8 trykknapper ("SW7 – SW0") på "Mega2560 I/O Shield" som "tangenter" på et klaver:



**SW. : 7 6 5 4 3 2 1 0**

De 8 trykknapper er fysisk forbundet til I/O porten PA på Mega2560.

Til brug for denne (del-) øvelse oprettes et nyt projekt (kald det "LAB4\_1") og der tages udgangspunkt i følgende kode (baseret på LAB3 løsningen):

```
;*****  
;***** MSYS LAB4,DEL 1 *****  
;***** Henning Hargaard *****  
;***** 7.september 2015 *****  
;*****  
;***** INITIERING *****  
LDI R16,HIGH(RAMEND) ;Initier Stack Pointer  
OUT SPH,R16  
LDI R16,LOW(RAMEND)  
OUT SPL,R16  
SER R16  
OUT DDRB,R16 ;PB benene er udgange  
  
;---> Skriv kode her, der konfigurer PA benene til INDGANGE  
  
;***** PROGRAM LOOP *****  
HERE:  
  
;---> Skriv kode her, der aftaster de 8 trykknapper og  
; spiller den korrekte tone, mens en knap holdes nede.  
; Når ingen knap holdes nede, skal lyd giveren være tavs.  
  
RJMP HERE  
  
;***** DELAY (R18*4us) *****  
DELAY:  
LDI R17,20  
AGAIN:|  
DEC R17  
BRNE AGAIN  
DEC R18  
BRNE DELAY  
RET  
;*****
```

Koden ovenfor er tilgængelig på MSYS Brightspace (filen hedder "LAB4\_1.asm").

Som det ses, mangler der kode, der konfigurer benene på "PA" til alle at være indgange (trykknapperne er tilkoblet denne port).  
Skriv denne kode, hvor det er indikeret med en pil.

Skriv nu koden, der udgør programmets hovedsløjfe (er også indikeret med en pil i koden).

Forslag til program-sekvens (med hints):

1. Start med at anvende 8 SBIS-instruktioner til at aflæse de 8 trykknapper en ad gangen. Hvis en given tast er trykket ned, loades registeret R18 med en værdi, der svarer til halvperiode-tiden ( $T/2$ ) for den til knappen hørende tone (se tabel 1 nedenunder).
2. Aflæs nu hele PINA-regiseret (anvend en IN-instruktion) for at finde ud af, om der er trykket nogen taster ned. Husk, at hvis ingen taster er trykket, aflæses der 0xFF fra PINA. Hvis der ikke er trykket nogen taster, skal der hoppes til label "HERE:". Gør brug af CPI-instruktionen efterfulgt af BREQ-instruktionen.
3. Toggle lyd giverens signal ved at: Invertere R16 (brug COM instruktionen til det) og sende R16 til PORTB (brug en OUT instruktion til det) .
4. Kald DELAY-funktionen (bemærk, at R18 allerede er loaded med "tonens  $T/2$ " under punkt 1).
5. Hop til label "HERE:".

Tone	Frekvens	$T = 1/f$	$T/2$	$(T/2)/(4\mu s)$
<b>c</b>	523,25 Hz	1911 $\mu s$	956 $\mu s$	239
<b>D</b>	587,33 Hz	1792 $\mu s$	851 $\mu s$	213
<b>E</b>	659,26 Hz	1517 $\mu s$	758 $\mu s$	190
<b>F</b>	698,46 Hz	1432 $\mu s$	716 $\mu s$	179
<b>G</b>	783,99 Hz	1276 $\mu s$	638 $\mu s$	160
<b>A</b>	880,00 Hz	1136 $\mu s$	568 $\mu s$	142
<b>H</b>	987,77 Hz	1012 $\mu s$	506 $\mu s$	127
<b>C</b>	1046,50 Hz	956 $\mu s$	478 $\mu s$	120

*Tabel 1: Frekvenserne + beregninger for en C dur skala.*

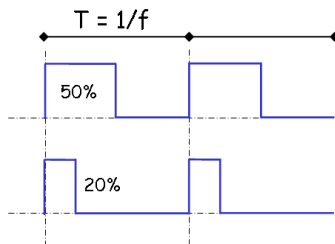
- Afprøv programmet på "Arduino Mega2560".
- Prøv at spille nogle (helst vellydende) melodier ved at trykke på "klaver-tasterne".

### Del 2 ("Lysdæmper")

I denne del af øvelsen, skal der skrives et program, som benytter et PWM signal til at styre lysintensiteten for en af "Mega2560 I/O Shield"s lysdioder.

Ved at justere på en lysdiodes procentvise ON-tid (duty cycle), vil lysdioden blinke med et tilsvarende mønster.

Hvis der anvendes en tilstrækkelig høj frekvens for signalet, vil øjet ikke opfatte, at lysdioden blinker, men i stedet opfatte, at lysintensiteten styres af duty cycle. Vi kan altså lave en simpel "lysdæmper".



Husk, at i vores hardware er trykknapperne forbundet til PA benene og lysdioderne til PB benene.

Til brug for denne del af øvelsen oprettes igen et nyt projekt (kald det "LAB4\_2"), og der tages udgangspunkt i følgende kode (ligger på Brightspace, filen hedder "LAB4\_2.asm"):

```
,*****
,***** MSYS, LAB4,DEL 2 *****
,***** Henning Hargaard *****
,***** 7.september 2015 *****
,*****
,***** INITIERING *****
,
LDI R16,HIGH(RAMEND)
OUT SPH,R16
LDI R16,LOW(RAMEND)
OUT SPL,R16

;---> Skriv kode her, der konfigurerer PB, ben 0 (LED0) til udgang
;---> Skriv kode her, der konfigurerer hele PA til indgange

,***** PROGRAM LOOP *****
HERE:

;---> Skriv kode her, der aftaster de 8 trykknapper og
;      styrer duty cycle på LED0 i henhold til
;      hvilken knap, der trykkes på

JMP HERE      ;Gentag loop

,***** DELAY (R18*4us) *****
DELAY:
LDI R17,20
AGAIN:
DEC R17
BRNE AGAIN
DEC R18
BRNE DELAY
RET
,*****
```

Som det ses, mangler der kode, der konfigurer benet PB, ben 0 til at være udgang (benyt SBI instruktionen). Desuden mangler der kode til at konfigurere "PA" benene til indgange. Skriv denne kode, hvor det er indikeret med en pil.

Skriv herefter koden, der udgør programmets hovedsløjfe.

Forslag til program-sekvens (med hints):

1. Start med at anvende 8 SBIS-instruktioner til at aflæse de 8 trykknapper en ad gangen. Hvis en given tast er trykket ned, loades registeret R20 med en værdi fra tabel 2 (se nedenfor).
2. Tænd lysdioden på PORTB ben 0.  
Benyt en single bit I/O-instruktion til det (se evt. side 149 i lærebogen).
3. Flyt (kopier) R20 over i R18 og kald DELAY funktionen.
4. Sluk lysdioden på PORTB ben 0.  
Benyt en single bit I/O-instruktion til det (se evt. side 149 i lærebogen).
5. Load R18 med tallet 128 og kald DELAY funktionen.
6. Hop til label "HERE:".

Switch	R20 loades med	Duty cycle
SW7	255	
SW6	128	
SW5	64	
SW4	32	
SW3	16	
SW2	8	
SW1	4	
SW0	2	

Tabel 2: PWM styring.

- Afprøv programmet.
- Tabel 2 indeholder en kolonne for duty cycle.  
Udfyld denne, når du ved, hvordan programmet fungerer.
- Hvad kan årsagen være til, at man har valgt en logaritmisk skale (og ikke en lineær skala) for ændringer af duty cycle?
- Prøv som et forsøg at ændre i programmets DELAY funktion, sådan at tidsforsinkelsen bliver meget større. Afprøv og konstater, at lysdioden faktisk blinker.